Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Лабораторная работа №4,5

**Информационные сети. Основы безопасности. Информационные сети. Основы безопасности.**

Тема:   
**«Разработка защищённых приложений»**.

Выполнил студент гр. 753504

Осипик И.Ф.

Проверил

Протько М.И.

Минск, 2020

# 1. Введение

Целью лабораторной работы является знакомство с концепцией ролевого управления доступом и способами защиты программного обеспечения от существующих угроз.

Научиться разрабатывать приложения, которые используют ролевое управление доступом для разграничения полномочий пользователей. Получить навыки защиты разработанной программы от несанкционированного копирования и других угроз, которым может подвергаться программное обеспечение.

**ЗАДАНИЕ:**

Реализовать программы, выполняющие указанные в задании действия.

**Задание №1.**

Реализовать приложение с графическим интерфейсом, удовлетворяющее следующим требованиям:

1. Приложение проводит аутентификацию пользователя.
2. Каждый пользователь программы должен относиться к какой-нибудь группе пользователей (роли), членам которой доступны различные функциональные возможности программы.
3. Программа должна принимать от пользователя некоторые данные и, возможно, после некоторой обработки, отображать их.
4. При этом должна осуществляться защита от как минимум 4-х типов возможных атак на приложение:

4.1) Атака «переполнение буфера».

4.2) Атака «SQL-инъекции».

4.3) Атака, эксплуатирующая ошибки канонизации.

4.4) Атака «XSS» (межсайтовое кодирование).

4.5) Принцип минимизации привилегий.

4.6) DoS-атака (отказ в обслуживании).

При разработке защиты нужно предположить, что приложение работает с базой данных, в которой сохраняет введённые пользователем данные.

**Задание №2.**

Реализовать приложение-инсталлятор, позволяющее установить на компьютер пользователя приложение, реализованное в предыдущем пункте задания.

Требования к приложению:

1. Приложение-инсталлятор совместно с устанавливаемым приложением должно обеспечивать защиту программного продукта от несанкционированного тиражирования.
2. Приложение-инсталлятор должно иметь защиту от возможных атак на него.

**Задание №3.**

Протестировать правильность работы систем защиты разработанных приложений посредством реализации тестовых атак выбранных 4-х типов.

# 2**. Блок-схемы алгоритмов**

# 3. Теоретические сведения

## **Атаки на сервер**

Переполнение буфера (buffer overflow) — наверное, одна из самых интересных и широко распространённых уязвимостей программного обеспечения. Ошибка заключается в том, что в каком-либо месте программы происходит копирование данных из одного участка памяти в другой без проверки того, достаточно ли для них места там, куда их копируют. Область памяти, куда копируются данные, принято называть буфером. Таким образом, если данных слишком много, то часть их попадает за границы буфера - происходит "переполнение буфера". Умелое использование того, куда попадают "лишние данные" может позволить злоумышленнику выполнить любой код на компьютере, где произошло переполнение.

Канонизация — это процесс, при котором все эквивалентные формы имени приводятся к одному стандартному имени — так называемому каноническому имени. Например, на определенной машине имена С: \dir\test. dat и . . \ . . \test. dat могут ссылаться на один и тот же файл. Канонизация — это процесс, во время которого все такие имена будут приведены к имени вида С: \dir\test. dat (по причине ошибок канонизации в сервере IIS) ... При запросе некоторых типов файлов с использованием специально подготовленного хакером адреса URL, канонизация приводит лишь к частично верному результату. Процедура канонизации дает верное имя файла, но приводит к выводу о том, что он расположен не в том каталоге, в котором он находится на самом деле. В результате для файла используются разрешения, установленные для другого каталога.

Принцип наименьших привилегий, (англ. Principle of least privilege) — принцип организации доступа к ресурсам, когда в тот или иной уровень абстракции от вычислительной среды, каждый модуль (такой, как процесс, пользователь или программа, которые мы рассматриваем) должны иметь доступ к такой информации и ресурсам, которые минимально необходимы для успешного выполнения его рабочей цели.

Это означает давать пользователю только те привилегии, которые являются абсолютно необходимыми для того, чтобы сделать свою работу. Например, пользователю, который занимается резервным копированием не нужно устанавливать любое другое программное обеспечение, кроме того, что ему необходимо для работы. Любые другие привилегии, например, как установка программного обеспечения и так далее должны быть заблокированы. Принцип распространяется также на пользователей персонального компьютера дома, где они должны работать в учетной записи обычного пользователя и открывать свою учетную запись администратора (защищённую паролем) только тогда, когда ситуация требует этого.

DoS (аббр. англ. Denial of Service «отказ в обслуживании») – затопление полуоткрытыми сессиями, переполняющими очереди сервера, после чего сервер перестает отвечать на запросы легитимных клиентов. Зачастую достаточно 50-100 ложных сессий и сервер будет «тормозить».

**Список литературы**.

1. Нортроп, Т. Разработка защищённых приложений на Visual Basic .NET и Visual C# .NET: учебный курс Microsoft / Т. Нортроп. – М: Издательство "Русская редакция", 2007. – 688 с.

2. Скляров, Д.В. Искусство защиты и взлома информации / Д.В. Скляров. – СПб: БХВ- Петербург, 2004. – 288 с.

3. Ховард, М. Защищённый код: пер. с англ. / М. Ховард, Д. Лебланк. – 2-е изд., испр. – М: Издательско-торговый дом "Русская редакция", 2004. – 704 с.

# 4. Результаты выполнения программы

# 5. Выводы

В данной лабораторной работе познакомились с концепцией ролевого управления доступом и способами защиты программного обеспечения от существующих угроз.

Приобрёл навыки в разработке приложений, которые используют ролевое управление доступом для разграничения полномочий пользователей.

Научился защищать программы от несанкционированного копирования и других угроз, которым может подвергаться программное обеспечение.

Научился реализовать программу, которая проводит аутентификацию пользователя, принимает от пользователя некоторые данные и, возможно, после некоторой обработки, отображать их, которая защищает от нескольких типов атак (Например, Атака «SQL-инъекции», Атака «переполнение буфера», DoS-атака и т. д.).

Научился реализовать приложение-инсталлятор, позволяющее установить на компьютер пользователя приложение, а также протестировал правильность работы систем защиты разработанных приложений посредством реализации тестовых атак.

# 6. КОД ПРОГРАММЫ